

# ESTRATEGIAS DIDÁCTICAS EN EL USO DE HERRAMIENTAS SOFTWARE PARA FAVORECER LA COMPRENSIÓN DE LOS ALUMNOS EN LA ENSEÑANZA SOBRE COMPILADORES

Gladis Sequeira, Silvia Zajackowski, Cristian Kornuta, Gabriela Gómez

Departamento de Informática/Facultad de Ciencias Exactas Químicas y Naturales/  
Universidad Nacional de Misiones

Felix de Azara 1552, Posadas, Misiones

Te: 376-422186

gladis.sequeira@gmail.com, silviazaj@yahoo.com.ar, cristian.kornuta@yahoo.com.ar, gabrielaasc@gmail.com

## Resumen

En este artículo se expone la investigación que se realiza dentro de la asignatura Teoría de la Computación de la carrera Licenciatura en Sistemas de Información, perteneciente a la Facultad de Ciencias Exactas, Químicas y Naturales de la Universidad Nacional de Misiones. El objetivo de la experiencia es constituir una propuesta didáctica curricular en torno a las prácticas realizadas con herramientas de software dentro de la cátedra, con el propósito de lograr una mejor comprensión y afianzamiento de los conocimientos en temas teóricos y obtener un prototipo funcional por parte de los alumnos al finalizar el cursado.

**Palabras clave:** Educación universitaria, Estrategias didácticas, Teoría de la Computación, Compiladores.

## Contexto

Este trabajo se enmarca en el “Programa de Investigación en Computación” del Centro de Investigación y Desarrollo Tecnológico de la Facultad de Ciencias Exactas Químicas y Naturales de la Universidad Nacional de Misiones.

Dentro del proyecto se desempeñan docentes, tesis, y becarios de las carreras de Analista en Sistemas de Computación, Licenciatura en sistemas de Información.

## Introducción

La planificación y ejecución de procesos de enseñanza-aprendizaje plantea un gran desafío a los docentes responsables de la cátedra Teoría de la Computación que se dicta en el primer cuatrimestre del tercer año de la carrera Licenciatura en Sistemas de Información de la Facultad de Ciencias Exactas, Químicas y Naturales perteneciente a la Universidad Nacional de Misiones. La tarea docente se complejiza a causa del grado de abstracción que requiere el cursado y la necesidad de relacionar todos los contenidos aprendidos en las cátedras anteriores de la carrera hasta ese momento.

El equipo docente, que se fue conformando a partir del año 2010, se abocó en su primer dictado a la enseñanza de los aspectos teóricos vinculados a los principios de diseño de compilación. Para la resolución de las guías prácticas se optó por aplicar el modelo “grupo cooperativo” organizándose, en esa oportunidad, diez grupos de trabajo al comenzar el cursado. Los prácticos consistieron en cuestionarios teóricos y construcción de mapas conceptuales integratorios. Al finalizar cada práctica, se realizaron la puesta en común del tema estudiado.

En dictados posteriores se incursionó en el desarrollo básico de prototipos de algoritmos, logrando que los alumnos trabajen en un proyecto donde implementaron un lenguaje de programación acotado tomando como referencia algún lenguaje imperativo aprendido

en cátedras anteriores. Así se logró un mejor afianzamiento de los contenidos teóricos.

A partir de los resultados obtenidos con el cursado del año 2011 la asignatura se planteó la necesidad de incluir herramientas informáticas, tales como software de simulación, que colaboren en el proceso de afianzar lo aprendido.

Como se expresa en [1], la posibilidad de probar un modelo, evaluarlo, ajustarlo y volverlo a probar hasta que el resultado sea satisfactorio contribuye al desarrollo de patrones o esquemas mentales adecuados para razonar y comprender. Así es posible que la aplicación de refinamiento sucesivo facilite al sistema cognitivo del sujeto la tarea de anticipar el comportamiento del modelo y permita redirigir ese esfuerzo hacia la comprensión de la teoría que respalda la práctica en proceso.

El software sería, en este caso, el elemento que simplificaría ese proceso de refinamiento sucesivo y ayudaría en la generación de relaciones pertinentes [1] que permitiese dotar de significado las expresiones regulares, autómatas finitos, gramática libre de contexto y árboles sintácticos que se emplean para describir el comportamiento de las distintas fases de un compilador [2] [3] [4].

## Objetivos

El objetivo de este proyecto es mejorar la comprensión de los alumnos dentro de la cátedra a partir de la incorporación de una estrategia didáctica que incluya herramientas software en las prácticas realizadas en la enseñanza de compiladores.

## Implementación de Simuladores y Metacompiladores para un Aprendizaje Constructivista

Nuestra experiencia nos llevó a proponer la utilización de simuladores, como ser el JFLAP [5] para la teoría de autómatas y el ANTRLWorks [6] [7] para el árbol sintáctico,

y la práctica, entendiéndose que los simuladores son solo herramientas de software que ayudarían al alumno a comprender los temas centrales de la cátedra.

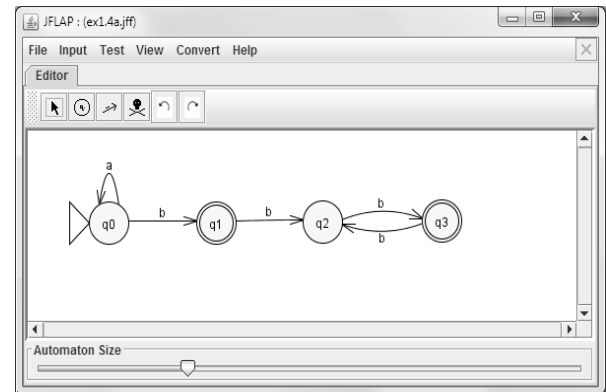


Fig. 1. JFLAP/ Autómatas finitos

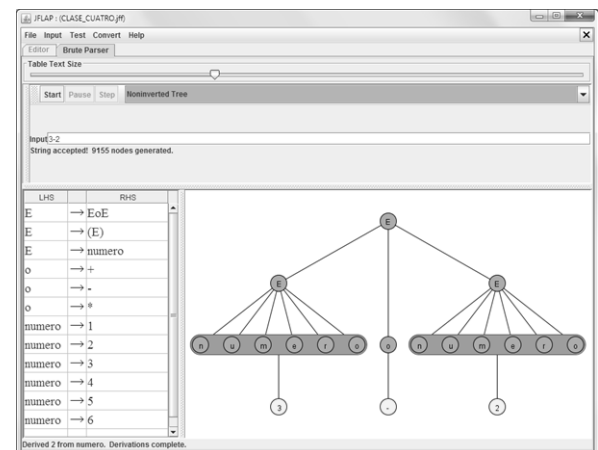


Fig. 2. JFLAP/ Arbol sintactico

Buscando estrategias didácticas que fortalezcan la comprensión de los alumnos y acompañen el Proceso Enseñanza-Aprendizaje dentro de la cátedra Teoría de la Computación, se optó por aplicar el enfoque constructivista. Este modelo define que el conocimiento se construye a partir de los esquemas que el individuo ya posee, sus conocimientos previos. Se trata de un proceso que se desarrolla día a día, centrado en la persona que aprende y que le permite adquirir, no solamente un nuevo conocimiento, sino también una nueva competencia que podrá aplicar a futuras situaciones [8].

Con esta intensidad se planteó a los alumnos, en la segunda etapa del cursado, el desarrollo de

un trabajo integrador que consistió en la implementación de un prototipo de lenguaje. El estudiante tenía dos posibles caminos a seguir, optar por el uso de un lenguaje de programación que conociera o utilizar algún metacompilador; en ambos casos obtendría soporte y respuestas a consultas referentes a la construcción de su propio lenguaje.

Si bien se presentaron las características generales de diferentes metacompiladores, como ser PLY (Python, Lex y Yacc) [9], ANTRL, JFlex y Bison, un relevamiento expuso que en cinco de los siete grupos que permanecían activos en el último mes de cursado, se encontraba al menos un alumno que tenía conocimiento de Python por haber realizado un curso de Extensión el año anterior. Esta situación decidió la presentación más detallada del metacompilador PLY que se caracteriza por implementarse en su totalidad en Python, además es sencillo de usar y proporciona una extensa comprobación de errores. PLY proporciona la base lex / yacc.

Cada grupo decidió, según sus conocimientos de programación, las herramientas de software a utilizar para implementar las primeras fases del prototipo avanzado de su lenguaje.

Durante esta etapa se continuó trabajando en forma grupal, con el modelo de “aprendizaje cooperativo”, considerando que la construcción del conocimiento, según el enfoque utilizado, se produce cuando el alumno interactúa con otros [8]. La consigna incluyó la presentación del producto obtenido.

## Líneas de investigación y desarrollo

Algunos de nuestros propósitos próximos son:

- Elaborar un Cuaderno de Cátedra con la finalidad de presentar al alumno una base teórica, guía de trabajo a seguir, ejercitaciones prácticas y cuestionarios.
- Conformar un soporte multimedial que permita contener al alumno en el transcurso del dictado de la cátedra y

que le ayude a profundizar y ampliar todos aquellos temas que se tratan en la cátedra.

- Diseñar e implementar alguna herramienta de programación visual, orientado a un lenguaje específico con propósitos educativos.
- Diseño e implementación de un Framework de trabajo orientado a un lenguaje específico o a la conjunción de las herramientas de desarrollo.

## Resultados y Objetivos

De los diez grupos cooperativos que se conformaron al iniciar el dictado 2012, tres abandonaron el cursado antes de iniciar el trabajo integrador. Si bien no se investigaron las causas de tal decisión, se presume que las razones podrían deberse a sobrecarga de actividades académicas y laborales.

Se realizó una encuesta anónima a fin de analizar la opinión de los estudiantes respecto a la incorporación de las herramientas JFLAP y ANTLRWorks en las prácticas correspondientes a las fases analizador léxico y sintáctico como apoyo de los conceptos teóricos. La misma arrojó como resultado que el 72% de los alumnos, sobre un total de 18, concluyeron que las prácticas realizadas con software le sirvieron para entender los conceptos teóricos de la asignatura.

Los siete grupos que completaron el cursado 2012 lograron desarrollar, con diferentes grados de complejidad, el prototipo solicitado. Cinco grupos trabajaron con la herramienta PLY y dos lo implementaron usando .NET.

## Conclusiones

La experiencia que implicó la incorporación de las herramientas de software dentro de la cátedra produjo buenos resultados para ambas partes involucradas en el proceso: los docentes se encontraron con actividades productivas realizadas por los estudiantes en sus prácticas, y los alumnos demostraron mejor destreza y

confianza en ellos mismos al momento de trabajar con las herramientas y confeccionar su propio prototipo de lenguaje de programación.

Por ser éste el primer dictado en que la asignatura implementa un trabajo integratorio con estas características, no existen datos previos que puedan servir de comparación, aunque el equipo de cátedra reconoce el avance logrado en cuanto al grado de comprensión de los contenidos e interés en las actividades planteadas, cuestiones que se vislumbran al observar la calidad de los prototipos diseñados y la originalidad de los mismos.

Una encuesta posterior al trabajo integrador arrojó que para el 82% de los estudiantes fue un desafío motivador para seguir investigando y les permitió integrar los conceptos teóricos dados en la cátedra.

## Formación de Recursos Humanos

En esta línea de investigación se está desarrollando una tesis de grado en la carrera Licenciatura en Sistemas de Información. Conjuntamente se prevé conformar un grupo de investigación en el marco del Departamento de Informática de la F.C.E.Q. y N. orientando las líneas de investigaciones futuras a la construcción de compiladores.

## Referencias

- [1]. Santos, G., Otero, M.R., Fanaro, M.A.: ¿Cómo usar software de simulación en clases de física?. Cuaderno Catarinense de Ensino de Física vol. 17 n.1 (2000)
- [2]. Kenneth, L.: Construcción de compiladores, principios y práctica. International Thomson Ed. (2004)
- [3]. Ruiz Catalan, J.: Compiladores TEORÍA E IMPLEMENTACIÓN. International Thomson Ed. (2010)
- [4]. Aho, A., Lam, M., Sethi, R., Ullman, J., Compilers, Principles, Techniques & Tools. Pearson Addison – Wesley (2008)
- [5]. JFlap, <http://www.jflap.org/>
- [6]. Parr, T.: The Definitive ANTLR reference. Raleigh (North Carolina) (2007)
- [7]. ANTLRWorks: The ANTLR GUI Development Environment, [wwwantlr.org/works/](http://wwwantlr.org/works/)
- [8]. Carretero, M.: Constructivismo y Educación. Aique (1993)
- [9]. PLY (Python Lex-Yacc), [http://www.dabeaz.com/ply/ply.html#ply\\_nn1](http://www.dabeaz.com/ply/ply.html#ply_nn1)